Local Exponential Maps: Towards Massively Distributed Multi-robot Mapping

Frank Dellaert, Alireza Fathi, Alex Cunningham, Balmanohar Paluri, and Kai Ni

***Abstract*— We present a novel paradigm for massively distributed, large-scale multi-robot mapping. Our goal is to explore techniques that can support continuous mapping over an indefinite amount of time. We argue that to scale to city or even global scales the concept of a single globally consistent map has to be abandoned, and present an infrastructure-supported solution where most of the inference and map-maintenance is done on local "map-servers", rather than on the robot itself. The main technical contribution in the paper is a factor-graph-based scheme for making this possible, and a novel local map representation, *local exponential maps*, that enable indefinite map updates while remaining self-consistent over time. We present initial experimental results both in simulation and using real data, although a full-scale deployment and evaluation of the technique is left for future work.**

I. Introduction

In this paper we present a novel paradigm for massively distributed, large-scale multi-robot mapping. Our goal is to explore techniques that can support continuous mapping over an indefinite amount of time. This is ultimately targeted at unbounded environments populated by a heterogeneous collections of robots with various capabilities and navigation needs. As argued convincingly in [1], to scale to city or even global scales the concept of a single globally consistent map has to be abandoned. In addition, we believe that such a scenario calls for an infrastructure-supported solution where most of the inference and map-maintenance is done on local "map-servers", rather than on the robot itself. The main technical contribution in the paper is a factor-graph-based scheme for making this possible, and a novel local map representation, *local exponential maps*, that enable indefinite map updates while remaining self-consistent over time.

Solutions for mapping and localization in a single global map have been explored in the SLAM literature for a long time now, since the seminal work by Smith and Cheeseman [2], Leonard and Durrant-Whyte [3], and others [4], [5]. An overview of the recent state of the art can be found in the articles by Durrant-Whyte and Bailey [6], [7] and the book by Thrun et al. [8]. Several authors have also explored multi-robot versions of these algorithms [9]–[12], where typically a single, shared map representation is used by all robots.

The idea of using multiple local maps has received a lot of traction in a single-robot context [1], [13]–[15], as it leads to more computationally efficient algorithms. In addition, as mentioned by Tardós et al. [14], local maps lend themselves naturally to multi-robot mapping, as strategies for map-merging can just as well serve to merge maps built by different robots. Several authors have exploited this idea and proposed true multi-map, multi-robot algorithms that have some very appealing properties [16]–[19].

This work was supported by the National Science Foundation

All authors are with the Georgia Institute of Technology School of Interactive Computing, 801 Atlantic Drive, Atlanta, GA 30332, USA

Corresponding author is Frank Dellaert, frank@cc.gatech.edu

A. Massively Distributed Robot Mapping

To scale to much larger scales, however, the concept of a single globally consistent map has to be abandoned. In the work cited above, typically the final goal is still to produce a global, merged map. An exception to the rule is the work by Bosse et al. [1], who argues convincingly that a collection of local, interconnected maps are a more scalable approach to large-scale mapping, regarding global map as a post-processing step, more for the benefits of human operators.

Hence, as our map representation *we use a set of local maps which collectively serve to describe the environment in which the robots move*, without any attempt to merge these maps or embed them into a global coordinate frame. The local maps are updated by any robot that traverses them, by the robot "donating" the information (literally) contained in its observations to the local local map which executes an efficient information space update. Simultaneously, the robot pose is marginalized out of the local local map and returned to the robot in a quid pro quo. By partitioning observations among local maps, no observation is double-counted, which is important to support visualization and verification by computing global representations.

This idea lends itself to a massively distributed implementation, as all map updates happen at the local level. The robots need not compute anything themselves: instead we opt for a scheme where *each local map performs the local updates and computes the pose for the robot.* This lends itself to interesting possibilities for large-scale, infrastructure-supported deployments, e.g., a collection of servers distributed throughout the environment that provide local maps for use by the robots to navigate locally and interact with the local environment. It might also be possible to efficiently store the local local maps in the environment itself, for example in later-generation RFIDs [20].

B. Local Exponential Maps

To support the scenario above we also introduce a novel local map representation, *local exponential maps*, that enables indefinite map updates while remaining self-consistent over time. In addition to being more efficient, local maps have also been shown to be beneficial in terms of reducing consistency problems with regards to linearization errors

[15], [21]. However, repeated linearization in EKF-style is not sustainable in the long-term.

Hence, in our implementation, the local maps keep a square-root information-form [22] estimate of the relative pose to local landmarks, but this estimate is kept in the tangent space $\mathfrak{g}$ of the Lie Group G that describes the local map configuration. The linearization is done once and only once, ensuring that all information added to the local map is using the same tangent space. A local estimate is then obtained through an exponential map from $\mathfrak{g}$ back to G.

C. Assumptions and Scope

We present proof-of-concept results, both with real and simulated data, that make a number of assumptions that are not realistic in all possible deployment scenarios. The results are meant to illustrate the main ideas in the paper and not as a fully-fledged demonstration of a distributed mapping system. The main assumptions we make is that robots are able to correctly recognize which local map(s) they are traversing, and that they are able to robustly estimate their local pose from a single measurement. We realize that this is a tall order, but others are working on exactly these problems, e.g., through local map matching [1], [23] or through robust appearance matching [24]. Another approach, which we adopt in our experiments, is by using a priori known and recognizable landmarks. Note that this can be a viable strategy in a number of realistic application scenarios: sometimes engineering oneself out of the unconstrained data association problem is a defensible choice.

We present experimental results with a single map-server that, in addition to the local maps it is able to serve to the robots, also keeps track of the topology of the local maps in order to facilitate planning over longer distance. Global plans are requested by the robots and computed by the server, after which a series of local map interactions is used to implement detailed navigation. More distributed planning schemes in sensor networks have been explored by others [25], [26] and could be used here as well.

II. Massively Distributed Robot Mapping

As discussed above, as our map representation we use a set of local maps M_i which collectively serve to describe the environment in which the robots move. Each local map has a coordinate frame with unknown 3D pose $p_i \in SE(3)$ in a global coordinate frame. However, but this pose is not actually computed as part of the mapping process. Instead, the map contains a state estimate of local map features and/or the relative location of other maps, in the form of a posterior probability density $P(M_{it}|Z^t)$ of the map M_{it} at time t, given all measurements $Z^t = \{z_i\}_{i=1}^t$ up to time t. Note the use of the subscript to denote a set of measurements, versus a single observation z_t at time t. Also, we assume that each local map has its own private time index t.

The local maps are updated by any robot that traverses them, by the robot "donating" the information contained in its observations to the local local map, which executes a local update. This step will be discussed in detail in Section II-A below. Simultaneously, the robot pose is marginalized out of the local local map and returned to the robot in a quid pro quo, as discussed in Section II-B.

A. Local Map Update

When a robot travels in the environment and it makes an observation z_t, the current local map M is updated. This is done by updating the posterior probability $P(M_t|Z^t)$ of the map M by incorporating the current observation z_t, and marginalizing out the local robot pose x_l:

$$\begin{aligned} P(M_t|Z^t) &= \int_{x_l} P(x_l, M_t|Z^t) \\ &\propto \int_{x_l} L(x_l, M_t; z_t) P(M_t|Z^{t-1}) \end{aligned} \quad (1)$$

Here we have only assumed that robots can make measurements z_t that induce a likelihood constraint $L(x_l, M; z) \propto P(z|x_l, M)$ between the local robot pose x_l and the local map M, where $P(z|x_l, M)$ is an a priori known measurement model. For static environments the predictive density $P(M_t|Z^{t-1})$ is assumed equal to the posterior from the previous update, i.e., $P(M_t|Z^{t-1}) = P(M_{t-1}|Z^{t-1})$.

Eliminating the local robot pose x_l is done by normalizing the likelihood constraint $L(x_l, M_t; z_t)$ to a conditional density $P(x_l|M_t; z_t)$ on the pose x_l:

$$P(x_l|M_t; z_t) = \frac{L(x_l, M_t; z_t)}{f(M_t; z_t)} \quad (2)$$

The normalizing factor

$$f(M_t; z_t) \triangleq \int_{x_l} L(x_l, M_t; z_t) \quad (3)$$

is a unary factor on M_t that encapsulates *all* the information contained in the measurement z_t about the local map M_t, and computing it is the main computational expense in the map update. Substituting (2) into (1) and simplifying, we obtain the posterior as the straightforward fusion of the information $f(M_t; z_t)$ contained in z_t on the one hand, and the predictive density $P(M_t|Z^{t-1})$ on the map M_t on the other hand:

$$P(M_t|Z^t) \propto L(M_t; z_t) P(M_t|Z^{t-1}) \quad (4)$$

The computation above is quite general and does not make a lot of assumptions about the nature of the landmarks. For example, in a visual SLAM scenario the local map can consist of a constellation of 3D features, in which case the map update step (4) is incremental structure from motion [27] and the pose recovery (5) is a probabilistic version of triangulation. On another note, our implementation is based on factor graphs, and the combination of steps (3) and (4) is nothing but a specialized version of the generalized sum-product algorithm to perform inference in factor graphs [28].

B. Pose Recovery

After the posterior $P(M_t|Z^t)$ on the local map M_t is computed, we can also recover a posterior $P(x_l|Z^t)$ on the

local pose x_l:

$$\begin{aligned} P(x_l|Z^t) &= \int_M P(x_l, M_t|Z^t) \\ &= \int_M P(x_l|M_t; z_t)P(M_t|Z^t) \end{aligned} \quad (5)$$

Hence, after updating a local map, the robot has an updated pose estimate at its disposal to perform planning and navigation. Note that odometry information was not discussed above, but it can be easily incorporated in both map update and pose recovery steps.

III. Local Exponential Maps

For the case of Gaussian measurement noise we introduce a novel local map representation, *local exponential maps*, that enables indefinite map updates while remaining self-consistent over time. Under the assumption of normally distributed measurement noise, the likelihood constraint in (1) is of the form

$$L(x_l, M_t; z_t) = \exp -\frac{1}{2}\|h(x_l, M_t) - z_t\|^2_\Sigma \quad (6)$$

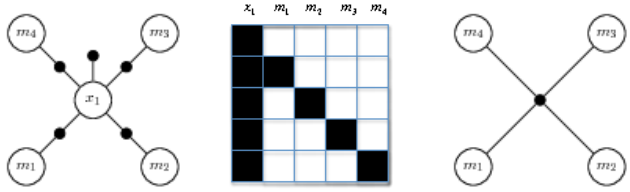
where h is typically a non-linear *measurement function*. In a typical EKF scenario we would proceed by linearizing (6) and fusing it into a Gaussian approximation of the prior, hence obtaining a Gaussian posterior which allows the algorithm to continue on recursively. However, an EKF will become inconsistent over time because the linearization at each step is done around the current posterior mean, and this will inevitably change over time, leading to inconsistency.

To remedy this, we keep the Gaussian density in square-root information-form [22], but define the density in the tangent space $\mathfrak{g}$ of the Lie Group G that describes the local map configuration. The linearization around a linearization point M_0 is done once and only once, ensuring that all information added to the local map is using the same tangent space. Any solution M can then be expressed as the exponential map of some point in the tangent space, i.e.,

$$M = M_0 \otimes exp(\delta M)$$

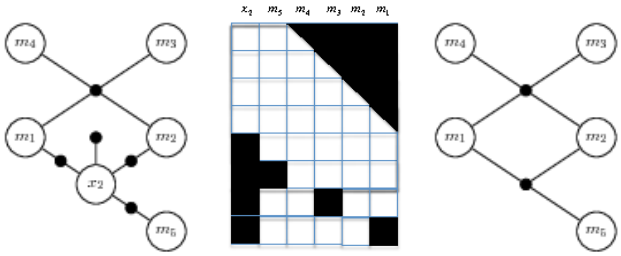
The local map at any given time is then the triple $(M_0, \delta M, R)$, where R is the upper triangular square root covariance. The factors are linearized around M_0 into measurements rows that are inserted into R using incremental QR factorization, as in Kaess et al. [29]. The incremental factorization updates are exact for linear factors, and hence if we never change the linearization point M_0, the square root information matrix R contains an *exact* record of all measurements incorporated so far, modulo the linearization.

Note that *the linearization point M_0 does not have to be at the global minimum*. For example, if the objective function was quadratic (linear case) you can put the linearization point anywhere -even very far away from the global minimum- and the solution would always be exact. Hence, the only requirement is that the linearization point M_0 is within the quadratic regime of the global minimum's basin of attraction. This can be ensured by, when a new local map is started, caching the first few measurements in a non-linear bootstrapping phase using a smoothing scheme [22], and only starting up the square root filter after the solution has sufficiently converged.



(a) (b) (c)

Figure 1: The factor graph for a local map when initially created. (a) The robot (x_1) observes four landmarks $m_1 ... m_4$, inducing four binary factors in addition to one unary factor which represents the observation of the reference landmark. (b) The corresponding measurement matrix A. (c) By eliminating the pose x_1, we obtain a single Gaussian factor which connects all landmarks. Elimination is equivalent to performing QR factorization on the matrix A [22].



(a) (b) (c)

Figure 2: Local map update explained using factor graphs. (a) The local map from Figure 1 is visited by a second robot (x_2) which adds 4 new observations to the local map, one of which is on the reference marker. (b) In matrix terms, the square root information matrix R is now augmented with four new measurement rows. (c) By eliminating x_2, we obtain a factor which is connected to all landmarks seen.

IV. Proof-of-Concept Implementation

A. Landmarks as Allocentric Reference Frames

In our proof-of-concept implementation we assume that the environment contains a set of recognizable landmarks L_i that can serve as allocentric coordinate frames. *Each landmark L_i, with pose $p_i \in SE(3)$, is identified with a local map M_i.* In this local map M_i in which we keep the the relative poses $r_{ji} \triangleq p_j \ominus p_i$ to other landmarks. Here we denote composition of poses by

$$p_i \otimes p_j = (R_i, t_j) \otimes (R_i, t_j) \triangleq (R_i R_j, t_i + R_i t_j)$$

and transforming one pose p_j into the frame of another landmark, L_i with pose p_i, by

$$r_{ji} \triangleq p_j \ominus p_i \triangleq p_i^{-1} \otimes p_j$$

with $p^{-1} \triangleq (R^T, -R^T t)$. Any robot likewise has a global pose $x_g \in SE(3)$, but at any given time one of the landmarks is picked as the reference landmark, say L_i, and hence the robot's pose can be described by a local pose $x_l \triangleq x_g \ominus p_i$.

As discussed, the main assumption we make is that a robot can unambiguously recognize a landmark, allowing a robot to always know in which local map(s) it is traveling. In addition, in our implementation we make one more assumption, namely that the robot's pose x_g can be robustly estimated relative to a single known landmark L_i. Formally, this means that the maximum likelihood estimate

$$\hat{x_g} = \underset{x}{\text{argmax}} \;\; L(x_g; p_i, z_i)$$

given an observation z_i on the landmark L_i with pose p_i is well defined, with no need for a prior $P(x_g)$ on the pose x_g.

In the local coordinate frame of the landmark M_i the likelihood estimation can be rephrased as maximizing a unary factor on the local robot pose x_l

$$\hat{x_l} = \underset{x}{\text{argmax}} \;\; g(x_l; z_i),$$

as we assume that the measurement z_i is a relative pose measurement and the relative pose of the landmark M is zero in its own coordinate frame, i.e.,

$$L(x_g, p_i; z_i) = L(x_l, r_i; z_i) = L(x_l, 0; z_i) \triangleq g(x_l; z_i)$$

B. Local Exponential Maps

The local maps now simply contain the relative pose r_j of all other landmarks L_j that were ever observed by robots while traveling in the local map M_i. The space of local map configurations, in the case of a map with n landmarks, is then the Lie group $SE(3)^n$, with the local tangent space $\mathfrak{se}(3)^n$. If at time t a robot observes a reference landmark L_i and n other landmarks M_j, the measurement constraint for the map update (1) then is the product of $n+1$ factors

$$L(x_l, M_t; z_t) = g(x_l; z_{ti}) \prod_{j=1}^{n} L(x_l, r_{tj}; z_{tj}) \qquad (7)$$

These can be used to update the map sequentially or in one calculation, using incremental QR factorization [29].

It is instructive to look at the local map update in terms of factor graph inference. Figures 1 and 2 respectively show the graph of factors corresponding to Equation 7 when the map is first created, and at a later time when a visit by a second robot adds a new landmark to the map. Also shown is the sparsity structure of the corresponding Jacobian A after linearization of the non-linear factors. Each column of A corresponds to one of the unknowns which are the nodes in the graph. Within each block-row, the sparsity pattern indicates which unknown robot or landmark poses are connected to the factor. Finally, eliminating variable nodes from the factor graph is equivalent to QR factorization in Figure 1, and incremental QR factorization in Figure 2.

The addition of new observation and optimization of the linear factor graph in each frame costs $O(|M|^2)$ where $|M|$ denotes the number of landmarks in the local map. Since the robot sees only a few landmarks in each local map, $|M|$ is bounded by a small constant number, which means that the addition of the new factors to the graph and optimization can be performed in constant time in each frame.

C. Client-Server Protocol

Our implementation is based on a client-server architecture, where local maps are served up by a server to the robots. A robot-client sends a message to the local map-server with its local observations, which selects the correct local map and updates it with the new information, as discussed in Section II-A. In the process, the marginalized-out pose (Section II-B) is sent back to the robot-client.

D. Global Optimization for Visualization

While a mobile robot can navigate perfectly using only local maps and planning in the graph of local maps, it is sometimes desired to create a global overview of all maps, perhaps for the benefit of an operator. To globally optimize the local maps, we put a constraint between every reference landmark L_i and the landmarks L_j seen in that reference landmark's local map. These factors implement constraints of the form

$$r_{ji} = p_j \ominus p_i \qquad (8)$$

which are the constraints arising from the same landmark being seen in two different local maps. We optimize the factor graph built from the constraints from all local maps, and get the global map. In practice we use soft constraints that penalize deviations from the equality constraints above quadratically, yielding a similar but smaller least-squares problem on the landmarks only. As an aside, by having partitioned observations among local maps, no observation is double-counted in the global optimization process.

V. Experimental Results

In our proof-of-concept experiments we use ARToolKit to detect fiducial markers [30] which we installed in an indoor environment. These markers have been used extensively in augmented reality applications and allow a robot to determine its relative pose with respect to the marker from a single view. In this case, the measurement z on a single marker is the set of pixel coordinates of the 4 corners detected in the image. The ARToolKit code also detects the unique marker ID based on its unique appearance. The simulations below use the same measurement model, and we generate synthetic markers with realistic sizes seen by virtual cameras with realistic camera parameters.

A. Simulation

Figures 3, 4, and 5 show simulation results for 50 synthetically generated planar landmarks in a 2D volume. In this case a single robot executed a regular traversal of the space (Figure 3), and at each of the 263 locations on its path sent a message to the map-server with its new observations. We used an accurate model of the visual measurement process, using a simulated 640*480 camera with a fairly wide focal length of 300 pixels, and simulated noise with a standard

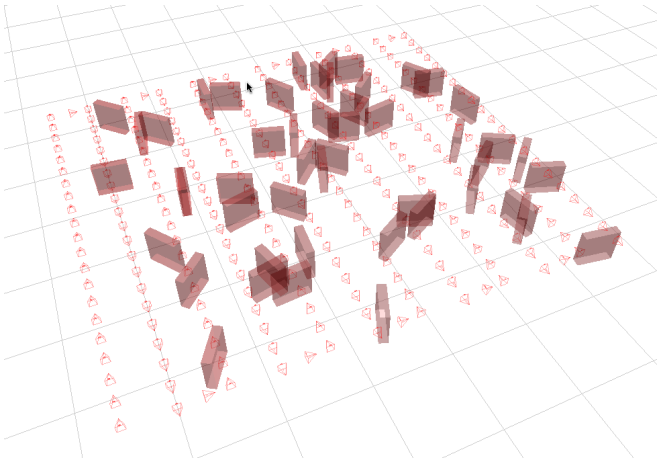


Figure 3: 2D simulation of a single robot traversing a set of randomly generated planar landmarks.

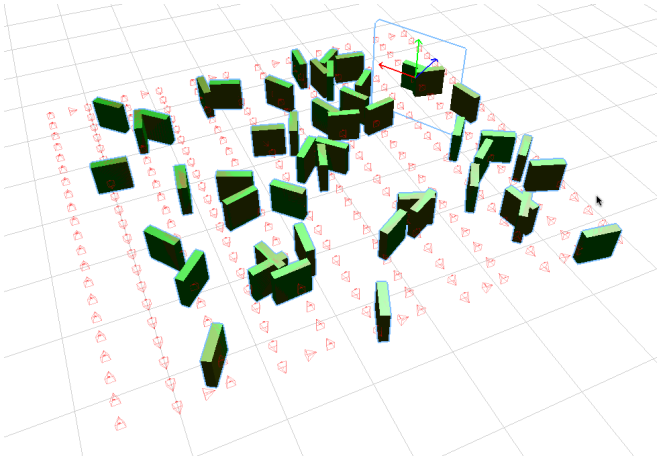


Figure 4: Result after gathering all local maps created by the simulated run from Figure 3 and globally optimizing the local map poses, as explained in Section IV-D.

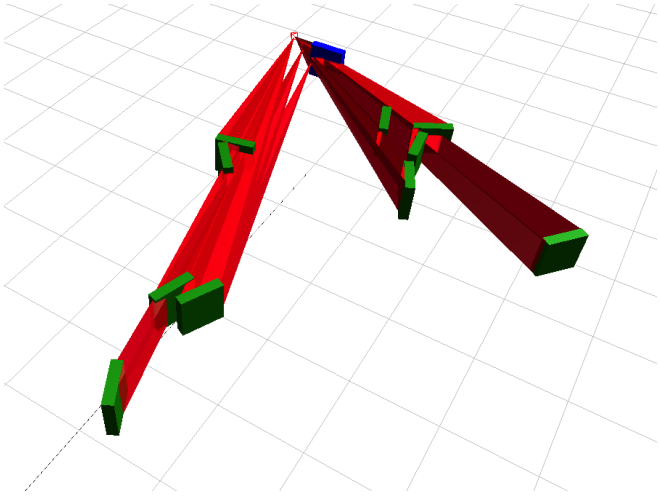


Figure 5: Local map for the landmark highlight in Figure 4. The red tetrahedrons indicate lines of sight. Note that we did not model occlusion in this simulation.

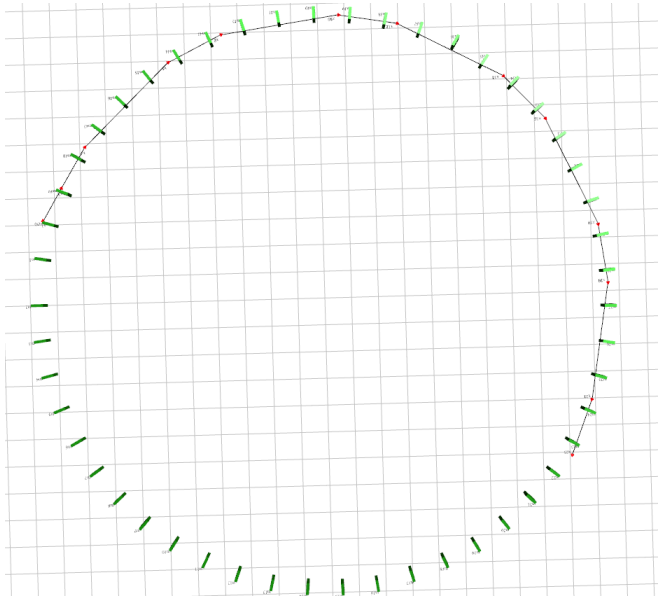


Figure 6: Mapping out a 1D sub-manifold through the use of interconnected local maps. Also shown is a path planned between two landmarks, using only the history of robot poses laid down previously.

deviation of 1 pixel. Occlusion was not modeled. In the process, 50 local maps were created, one for each landmark, and Figure 4 shows the visualization after post-processing for display in a single coordinate frame (Section IV-D). One of the local maps generated is shown in Figure 5.

B. Mapping Manifolds

The local, inter-connected nature of the local maps can be used to map manifolds in 3-space as well, similar to the ideas put forth in [18]. Figure 6 shows a simulation of a robot living in a 1D sub-manifold, traversing a set of landmarks. The example shows that the local maps accurately capture the 1D manifold structure when embedded in a global 3D coordinate frame. A less trivial application would be the manifold obtained by a robot traversing multiple floors in a building, which is a 2D manifold in 3-space. Note that we would be able to visualize such a manifold in 3-space, but that it is fairly meaningless to the robot: it only sees the local 2D manifold.

C. Navigation

The set of local maps created can be used as the basis for navigation by storing local navigation records within them. Using these embedded navigation histories, a robot passing through a local map to reach a destination can simply conduct non-parametric navigation by sampling among the stored navigation records. An important advantage of such non-parametric navigation is that the recorded trajectories implicitly deliver semantic constraints in the environments, such as maximum allowable speed or imposed navigation rules (move close to the walls). In particular, every robot that passes through a local map can leave its successful navigation

Figure 7: One of the local maps from the indoor experiment. The map is shown in orthographic projection over a blueprint of the actual environment.

paths in the local map and the accumulated paths can be used by the other robots in the future. Figure 6 shows a simple example where the robot plans a path from one marker to another in the 1D manifold.

D. Multi-robot Experiments

We conducted an indoor mapping experiment using the ARToolkit markers as the recognizable landmarks that can deliver robust pose. We recognize that these easily recognizable black-and-white markers might not be applicable in many applications. Our experiment was designed to demonstrate the feasibility of the approach in a real environment, with real visual measurements, and the use of the markers allowed us to concentrate on the mapping aspect.

In the experiments, 6 robot runs were executed, creating a total of 31 local maps over an area of about 1000 m^2. The different robot runs generated a total of 2308 robot to map-server interactions, respectively 627, 259, 514, 258, 146, and 504 for each of the different robot runs. Figure 7 shows one of the local maps that was created in the process. In this case, the local map ran in non-linear "bootstrap" mode until it switched to linear model (using the local exponential map method from Section III) after having received 46 measurements. The globally aligned visualization is shown in Figure 8.

VI. Discussion

We have presented a distributed multi-robot mapping framework, using a novel local map representation that operates in the tangent space of the local Lie group representation of the maps. An interesting aspect of the local map approach is that they can be used to map out complex sub-manifolds in 3D-space, which are ultimately more useful to robots than the nice 3D visualizations one can obtained by performing the global alignment of local maps.

Our proof-of-concept simulation and experiments presented here are just that: we make no pretense that we have demonstrated a realistic deployment, but do believe that we demonstrated the feasibility of the approach. In future work we plan a deployment on the scale of a university campus over a period of a month, to gain experience with the practical aspects of the approach and suggest ways to improve upon it. In those experiments we will most likely make use of image-based landmarks as in [31]. We are also very impressed by the work by Agrawal and Konolige on view-based perception for SLAM [32] which might be applicable here.

We believe that the novel, distributed multi-robot mapping framework presented here, building on the work by Bosse et al. [1], has all the right ingredients to be deployed on a massive scale. If our society will be revolutionized by robots, the most probable future scenario involves a lot of infrastructure support to enable robots with different capabilities to navigate and operate on our streets, within our cities, and inside our buildings. Just like desktop-processing is being off-loaded in the "cloud", we envision a future where it is not the robots doing the heavy lifting (in terms of mapping!), but where inference is done on a collection of locally accessible map servers that serve as a common substrate for navigation. Note that such an infrastructure is not limited to use by robots, but anyone with a mobile computing device might be able to benefit *and* contribute to a global mapping solution.

Acknowledgments

The authors would like to thank M.Sc. students Archana Asokan and Eohan George for their invaluable help in the experiments, and Sang Min Oh for his earlier contributions to the navigation part. In addition, *peekabot* was an invaluable tool in visualizing and debugging our code. Thanks!

References

[1] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework," *Intl. J. of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, Dec 2004.

[2] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Intl. J. of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1987.

[3] J. Leonard, H. Durrant-Whyte, and I. Cox, "Dynamic map building for an autonomous mobile robot," *Intl. J. of Robotics Research*, vol. 11, no. 4, pp. 286–289, 1992.

[4] N. Ayache and O. Faugeras, "Building, registering and fusing noisy visual maps," *Intl. J. of Robotics Research*, 1988.

[5] P. Moutarlier and R. Chatila, "Stochastic multisensor data fusion for mobile robot location and environment modelling," in *5th Int. Symp. Robotics Research*, 1989.

[6] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *Robotics & Automation Magazine*, Jun 2006.

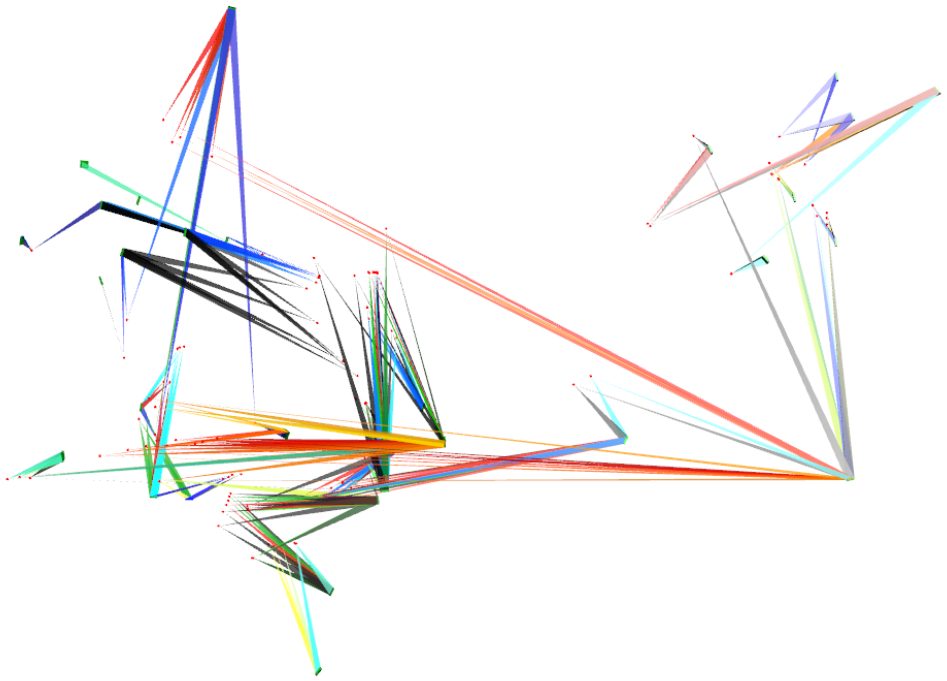


Figure 8: All the local maps visualized in a single coordinate frame which spans about 1000 m^2. The visibility polygons map out lines of sight in the environment, and are colored according to the local map index.

[7] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II state of the art," *Robotics & Automation Magazine*, Sep 2006.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.

[9] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *Intl. J. of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

[10] J. Fenwick, P. Newman, and J. Leonard, "Cooperative concurrent mapping and localization," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2002, pp. 1810–1817.

[11] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robotics*, 2005.

[12] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multi-robot exploration and mapping," *Proceedings of the IEEE - Special Issue on Multi-robot Systems*, vol. 94, no. 9, Jul 2006.

[13] K. Chong and L. Kleeman, "Large scale sonarray mapping using multiple connected local maps," in *Field and Service Robotics (FSR)*, 1997.

[14] J. Tardós, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Intl. J. of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.

[15] J. Castellanos, R. Martínez-Cantín, J. Tardós, and N. J., "Robocentric map joining: Improving the consistency of EKF-SLAM," *Journal of Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 21–29, January 2007.

[16] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "Towards multi-vehicle simultaneous localisation and mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2002.

[17] D. Rodriguez-Losada, F. Matia, and A. Jimenez, "Local maps fusion for real time multirobot indoor simultaneous localization and mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2004, pp. 1308–1313.

[18] A. Howard, G. S. Sukhatme, and M. J. Matarić, "Multi-robot mapping using manifold representations," *Proceedings of the IEEE - Special Issue on Multi-robot Systems*, vol. 94, no. 9, pp. 1360 – 1369, Jul 2006.

[19] M. Pfingsthorn, B. Slamet, and A. Visser, "A scalable hybrid multi-robot SLAM method for highly detailed maps," in *RoboCup 2007: Robot Soccer World Cup XI*, ser. Lecture Notes In AI, U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, Eds. Springer-Verlag, July 2008, pp. 457–464.

[20] M. S. B. Ransford, S. S. Clark and K. Fu, "Getting things done on computational RFIDs with energy-aware checkpointing and voltage-aware scheduling."

[21] S. Julier and J. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 4, 2001, pp. 4238–4243.

[22] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec 2006.

[23] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1999, pp. 318–325.

[24] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *Intl. J. of Robotics Research*, vol. 27, no. 6, pp. 647–665, June 2008.

[25] M. Batalin, G. S. Sukhatme, and M. Hattig, "Mobile robot navigation using a sensor network," *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 636–642, April 2004.

[26] K. O'Hara, D. Walker, and T. Balch, "Physical path planning using a pervasive embedded network," *IEEE Trans. Robotics*, vol. 24, no. 3, pp. 741–746, June 2008.

[27] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun 2007.

[28] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, February 2001.

[29] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec 2008.

[30] D. Wagner and D. Schmalsteig, "Artoolkitplus for pose tracking on mobile devices," in *Computer Vision Winter Workshop*, 2007.

[31] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[32] M. Agrawal and K. Konolige, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1066–1077, October 2008.